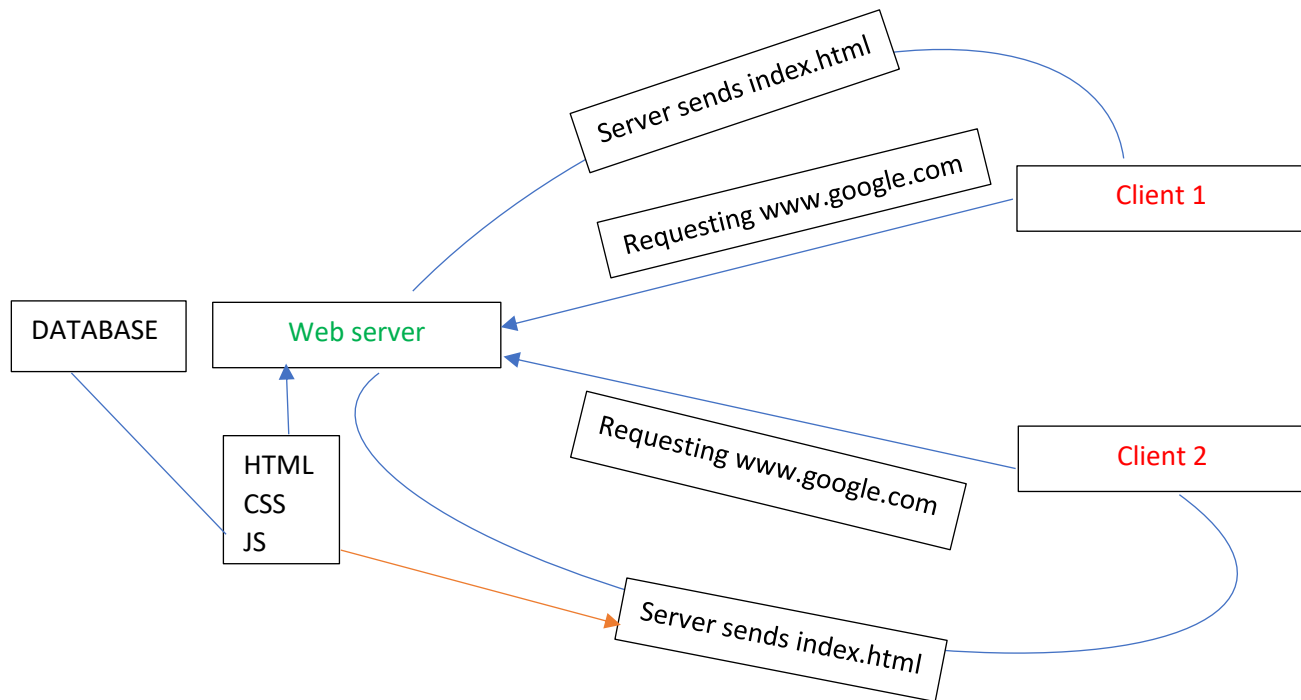


WEB DEVELOPMENT LEARNING CURVE

Custom site development involves a steep learning curve.



Index.html has HTML, CSS, JS combined and it will be send from server and browser will pass it to the client. How is the HTML, CSS, JS combination generated? Its done by the backend. Backend reads from database or file to generate it. Backend decides which response to send to which request. It can query blogpost, images etc.

Jumping to the learning curve. Before starting, its always nice to have someone who have been through it and can mentor In this path, otherwise with smallest of issues you need to jump to stack overflow. Stack overflow is great but in my personal experience one needs to understand the reason behind the problem as well, otherwise the second time the error come sin, you wont understand. In such cases a mentor is like a blessing. When I started with Springboot, having no mentor to guide, I found it extremely tough. Setup itself was an ordeal, followed by errors which I could not reason. With competitive coding at my back, I knew to debug and solve issues there, the same isn't the case with webdev. I struggled even to read the error, let alone understanding. Taking courses on Udemy might help, but based on personal experience, the courses there are outdated and instructors take time to respond. Issues become greater when you are following blindly what they are showing in their environment and suddenly you see a mismatch between yours and their output. So its best to tutor under a team who can guide in this learning curve.

Learning curve(At best 3 months minimum):

1. Frontend basic: 1. HTML 2. CSS 3. JS→ TypeScript.
2. GIT and GitHub version control system.
3. React or framework like Angular.
4. State management: Services(Angular).
5. So which backend language to take for absolute beginner?
 1. PHP: It is easy to learn, closest to JS. But libraries are less. [LARAVEL framework] **DATABASE: MySQL.**
 2. NodeJS: JS we already know by now. But JS complicated concept are involved here. [Express framework]. **DATABASE: MongoDB**
 3. Python: Django/ FLASK are the 2 backend frameworks. **DATABASE: SQLITE.**
 4. SpringBoot: I used Java to code and got to know that GE REN uses SpringBoot for backend, so before joining as a heads up, I tried preparing SpringBoot. Before SpringBoot you need to know Spring Framework, else dependency injection and other features wont come with clarity.
6. Relational DATABASE:
 1. MySQL.
 2. PostgreSQL
7. NoSQL DATABASE:
 1. MongoDB.
8. Application Programming interface.
9. Traditional and Headless CMS: WordPress.
10. Deployment/devops:
 1. SSH.
 2. SSL certificate.
 3. GitHub Actions.
 4. Load balancing, Monitoring, Security.
11. Static hosting: Netify or GitHub pages.
12. Web servers: Apache.
13. Testing: Unit, Integration and Functional.
14. Virtualisation: Automate container deployment, scaling and management: Kubernetes.

Note: I don't know most of the topics I mentioned above. Having researched on the roadmap I should take up to get ready as quickly I can for the website I needed to make for cash automation, I got to know in detail what I should pick up. Again, under a team's guidance one can do it in 3 months, without anyone to mentor, it will take much more time, specially if it's the first time you are doing web development.